การเรียงลำดับอักษรภาษาไทยด้วย dBASE III

นายแพทย์ ชุษณะ มะกรสาร*

ในการใช้งานและเขียนโปรแกรมภาษาไทย โดยใช้ดีเบสทรีนั้น ผู้เขียนได้รับคำถามอยู่เสมอ และหนึ่งในคำถามที่ถูกถามบ่อยที่สุดคือ มีทางใดไหมที่จะเรียงอักษร (sort) ภาษาไทยให้ถูกต้องได้ คำตอบคือทำได้ แต่ออกจะยุ่งยากและใช้เวลาสักหน่อย โดยเฉพาะถ้าใช้ภาษาดีเบสทรีล้วนๆ อันที่จริงเรื่องการเรียงลำดับอักษรนี้ ได้เคยมีผู้เขียนลงวารสารคอมพิวเตอร์ภาษาไทยบ้างแล้ว แต่ยังไม่สมบูรณ์นัก และคงจะใช้เวลาในการเรียงอย่างมาก ผู้เขียนเองก็มีความจำเป็นต้องเรียงรายชื่อผู้ป่วยคลินิคเบาหวานของโรงพยาบาลราชวิถีเอาไว้เผื่อค้นหาบัตรกรณีที่ผู้ป่วยลืมเอาบัตรมา ดังนั้นเร็วๆนี้จึงได้เขียนรูทีนสำหรับเรียงอักษรภาษาไทยขึ้นมา โดยใช้ภาษาแอสเซมบลีเข้าร่วมด้วย ทำให้การเรียงอักษรนี้เป็นไปได้รวดเร็วขึ้นมาก และได้แก้ไขให้การเรียงลำดับนี้ถูกต้องมากที่สุดด้วย และเมื่อเขียนขึ้นมาแล้วก็คิดว่าควรจะเผยแพร่ให้ผู้อื่นในวงการได้ใช้ประโยชน์บ้าง จึงได้เขียนบทความนี้ขึ้นมา ทั้งที่ผู้เขียนเองไม่ใช่นักเขียน และรู้สึกว่าไม่ค่อยจะถูกเส้นกับงานเขียนเท่าไรนัก

บทความนี้ ผู้เขียนใช้รหัสภาษาไทยของมหาวิทยาลัยเกษตรศาสตร์ (KU) โดยที่รหัสนี้นับว่าออกแบบมาได้ดี เพราะคำนึงถึงหลักใหญ่ๆ 2 ประการ คือ การจัดระดับตัวอักษร และ การเรียงลำดับตัวอักษร ทำให้ใช้งานได้ง่าย เขียนโปรแกรมจัด 3 ระดับได้ง่าย และเรียงคำได้ง่ายกว่ารหัสอื่นๆ นอกจากนี้ยังมีปัญหาเรื่องฮาร์ดแวร์น้อยกว่ารหัสอื่น เพราะจัดสระผสมเอาไว้แล้ว (รหัสอื่น แม้ใช้รหัสเดียวกัน อาจใช้เครื่องพิมพ์ หรือจอมอนิเตอร์คนละบริษัทไม่ได้ เพราะรหัสสระผสมไม่ตรงกัน) ถ้าท่านต้องการเรียงอักษรในรหัสอื่น คงต้องแก้ไขโปรแกรมเอาเองนะครับ สำหรับรหัสภาษาไทยของ KU แสดงไว้ในรูปที่ 1 เพื่อให้ติดตามบทความนี้ได้สะดวกยิ่งขึ้น

การเรียงอักษร ถ้าเป็นภาษาอังกฤษนั้น ทำได้ง่ายมาก เพราะมีรูทีนอยู่แล้ว โดยลักษณะของภาษาอังกฤษจะตรงไปตรงมา คือเรียงตามรหัสแอสกี โดยอักษรตัวแรกของคำถือว่ามีน้ำหนักมากที่สุดและลดหลั่นกันลงมา และแอสกีน้อยถือว่าน้ำหนักมาก ให้เรียงเอาไว้

รูปที่ 1 ตารางรหัสภาษาไทย มหาวิทยาลัยเกษตรศาสตร์

161	ก	162	ข	163	ค	164	ฆ	165	ง	166	จ	167	ฉ	168	ช
169	ซ	170	ฌ	171	ญ	172	ฎ	173	ฏ	174	ฐ	175	ฑ	176	ฒ
177	ณ	178	ด	179	ต	180	ถ	181	ท	182	ธ	183	น	184	บ
185	ป	186	ผ	187	ฝ	188	พ	189	ฟ	190	ภ	191	ม	192	ย
193	ร	194	ฤ	195	ล	196	ว	197	ศ	198	ษ	199	ส	200	ห
201	ฬ	202	อ	203	ฮ	204	ะ	205		206	า	207	ำ	208	เ
209	แ	210	โ	211	ใ	212	ไ	213	ๆ	214	ฯ	215	ุ	216	ู
217	ิ	218	ี	219	ึ	220	ื	221	ั	222	่	223	็	224	่
225	้	226	๊	227	๋	228	์	229		230	๋	231	๋	232	๊
233	๋	234	ั	235	ั	236	๊	237	๋	238	ิ	239	ี	240	๊
241	ิ	242	ิ	243	ี	244	ี	245	ี	246	ี	247	ี	248	ี
249	ี	250	ี	251	ี	252	ี	253	ี	254	ี				

* ชมรมไมโครคอมพิวเตอร์ โรงพยาบาลราชวิถี

ก่อน แต่วิธีดังกล่าวถ้ามาใช้ในการเรียงลำดับภาษาไทยแล้วจะมีปัญหาขึ้นทันที ลองดูตัวอย่างชุดรายชื่อต่อไปนี้

ชัยรัตน์ ชาตรี ชุมพร

ชุ่ม ชุลีพร ชีวิน

โชคชัย ดิเรก ดิศักดิ์

ถ้ามีการเรียงลำดับโดยใช้รูทีนในภาษาอังกฤษจะเรียงลำดับได้ดังนี้

ชาตรี ชีวิน ชุมพร

ชุลีพร ชุ่ม ชัยรัตน์

ดิศักดิ์ ดิเรก โชคชัย

จะเห็นความผิดเพี้ยนไปจากที่ควรจะเป็นมากทีเดียว ทั้งนี้ด้วยเหตุหลายประการคือ

1. สระหน้า ได้แก่ เ แ โ ไ ใ สระเหล่านี้ ทำให้น้ำหนักของคำลดลง จึงไปจัดเรียงอยู่ท้ายสุด เช่น ชื่อ โชคชัย จัดไปอยู่ท้ายสุด ชื่อ ดิเรก อยู่หลังดิศักดิ์

2. ั ตามตารางรหัสอยู่หลัง ็ การที่ KU จัดไว้เช่นนี้เพื่อง่ายต่อการจัด 3 ระดับ แต่ทำให้การเรียงอักษรผิดพลาด

3. วรรณยุกต์ ได้แก่ ่ ้ ๊ ๋ + เมื่อแทรกอยู่ในคำใด จะทำให้คำนั้นมีน้ำหนักลดลง ดังตัวอย่างจะเห็นว่า ชื่อ ชุ่ม จะจัดอยู่หลัง ชุลีพร

แนวความคิดในการจัดเรียงลำดับอักษรภาษาไทย

เอาละครับ ได้ตัวปัญหามาแล้ว เราลองมาดูแนวความคิดในการจัดเรียงลำดับอักษรให้ถูกต้องดูว่าจะทำได้อย่างไร

ถ้าเราเก็บชื่อที่ต้องการเรียงลำดับอักษรดังกล่าวเอาไว้ในฟิลด์หนึ่ง ของไฟล์ในดีเบสทรี จากนั้นทำการเปลี่ยนแปลงตัวอักษรในฟิลด์นั้นโดยใช้หลักข้างต้น ทำการจัดให้น้ำหนักของรหัสแอสกีเป็นไปตามการเรียงคำถูกต้อง (ดังจะได้กล่าวต่อไป) แล้วจึงทำการอินเด็กซ์โดยใช้ยูทิลิตี้อินเด็กซ์ของดีเบส เราจะได้อินเด็กซ์ไฟล์ซึ่งบ่งถึงการเรียงลำดับอักษรที่ถูกต้อง ในที่นี้จะขอกล่าวถึงการแก้ไขอักษรเพื่อให้มีการเรียงรหัสแอสกีที่ถูกต้องก่อนการอินเด็กซ์

การจัดเรียงน้ำหนักตัวอักษร

การจัดเรียงน้ำหนักตัวอักษรนี้ เป็นจุดสำคัญของการเรียงคำในภาษาไทยที่เรากำลังต้องการ สิ่งที่เราจะทำก็คือ ในเมื่อภาษาไทยตามรูปแบบปรกตินี้ เมื่อใช้อินเด็กซ์แบบภาษาอังกฤษแล้วเกิดการผิดเพี้ยนขึ้น เราก็แก้ไขโดยแก้ภาษาไทยของเราเสียก่อนทำการอินเด็กซ์ การแก้ไขนี้ก็เป็นการแก้ไขโดยอาศัยหลักการที่กล่าวมาแล้วข้างต้นดังต่อไปนี้

1. ถ้า <field> มีสระ เ แ โ ไ ใ ให้ทำการสลับที่ สระดังกล่าวกับ อักษรตัวถัดไป ทั้งนี้เพื่อลดน้ำหนัก เ แ โ ไ ใ ลง ลองดูชื่อต่อไปนี้

ชีวิน โชคชัย ดิศักดิ์

ดิเรก เมื่อสลับแล้วจะได้

ชีวิน ชโคชัย ดิศักดิ์

ดิรเก

จะเห็นว่าน้ำหนักของ ชโคชัย ก็ยังคงมากกว่า ชีวิน ยังคงไม่ถูกต้องสมบูรณ์ ดังนั้นเราต้องลดน้ำหนัก โ ลงไปอีก โดยเพิ่มแอสกีขึ้นอีกสัก 20 (รหัสเดิม โ=210 ให้เปลี่ยนเป็น 230 ถ้าเป็น เ ก็เปลี่ยนเป็น 208+20=228 เป็นต้น) ขอใช้สัญลักษณ์แทน เ แ โ ไ ใ ที่เปลี่ยนไปเป็น I II III IV V เพื่อง่ายแก่ความเข้าใจ

2. ั มีน้ำหนักไม่ถูกต้อง ให้ทำการเปลี่ยนให้เป็นรหัส 205 ซึ่งอยู่ระหว่าง ะ และ า ก็จะได้น้ำหนักที่ถูกต้อง ในที่นี้ขอใช้สัญลักษณ์แทนรหัส 205 ว่า ฮ

3. วรรณยุกต์ ให้ทำการสลับที่วรรณยุกต์กับอักษรถัดไปเพื่อลดน้ำหนักของวรรณยุกต์ลง

4. สุดท้ายคือ โปรแกรมควรตรวจสอบลำดับการพิมพ์ ว่าพิมพ์ถูกต้องหรือไม่ และจัดลำดับให้ถูกต้องด้วย เช่น

รูปที่ 2 โปรแกรม Thaimod.asm

```
;************************
;* listing THAIMOD.ASM *
;************************
; by Choosna Makarasara MD.
; Rajavithi Hospital Micro Computer Club
; 6 February 1988
; this program will open DBF file
; and modify field containing Thai character
; and adjust weight for using with dBASE index utility

code segment public 'code'
readf  equ 3fh
openf  equ 3dh
closef equ 3eh
writef equ 40h
movep  equ 42h
creaf  equ 3ch
       assume cs:code,da:data
       mov ax,ds
       mov es,ax
       mov ax,data
       mov ds,ax
       mov bx,80h
       mov al,es:[bx]          ;check parameter from input line
       or al,al                ;if no parameter
       jnz ck1
ck0:   mov ah,4ch              ;exit
       int 21h
ck1:   inc bx                  ;next char. from input line
       mov al,es:[bx]
       cmp al,' '              ;check space
```

ชื่อ ชุ่ม ผู้พิมพ์อาจพิมพ์ ช ุ ่ม หรือ ช ่ ุม ก็ได้ ดังนั้นต้องจัดการสลับที่ให้ถูกต้อง

อักษรที่มีปัญหาอีกตัวคือ ั ซึ่งในดีเบสทรีภาษาไทยมักเก็บเป็นรหัสผสม (รหัส 235) ดังนั้น โปรแกรมจะต้องแยกรหัสนี้ออกเป็น ั ั แยกจากกันด้วย

หลังจากทำตามลำดับทั้ง 4 ขั้นตอนนี้แล้ว จะได้ผลออกมาดังนี้

ชะยระะตน์ ชาตร์ ช ุมพร

ช ุม ่ ช ุล์พร ช์ว์น

ช็โคชะย ด์ริก ด์ศะกด์

ลองพิจารณาดูการเรียงรหัสแอสกี หลังจากการเปลี่ยนแปลงแล้ว จะเห็นได้ว่าน้ำหนักต่างๆ ถูกต้องสมบูรณ์แล้ว ถ้าใช้อินเด็กซ์ในดีเบสทรีจัดการเรียงลำดับอักษรดังกล่าวข้างต้นก็จะเป็นการเรียงที่ถูกต้องตามแบบภาษาไทย

คราวนี้ลองดูขั้นตอนทั้งหมดดังต่อไปนี้นะครับ สมมุติว่ามีไฟล์ DBF ที่เป็นต้นแบบให้ชื่อในที่นี้ว่า <source> และมีฟิลด์ที่ต้องการเรียงลำดับให้ชื่อในที่นี้ว่า <field> โดยเมื่อเรียงลำดับแล้ว

รูปที่ 2

```
        jz ck1                      ;if space - skip to next char.
        cmp al,13                   ;if CR
        jz ck0                      ;no string found - exit
        mov si,offset dbfname       ;save first string as dbfname
ck2:    mov al,es:[bx]
        cmp al,32                   ;if space process next string
        jz ck3
        cmp al,13                   ;if CR
        jz ck0                     ;only 1 string found - exit (require 2 string)
        mov [si],al
        inc bx
        inc si
        jmp ck2
ck3:    mov byte ptr [si],0         ;nul terminate for dbfname
        mov si,offset field         ;save second string as field name
ck5:    mov al,es:[bx]
        cmp al,13                   ;CR - end of input line - no second parameter
        jz ck0                      ;exit
        cmp al,32                   ;space - skip
        jnz ck4                     ;not space - second parameter found
        inc bx
        jmp ck5
ck4:    mov al,es:[bx]
        cmp al,13
        jz ck6
        cmp al,32
        jz ck6
        mov [si],al
        inc bx
        inc si
        jmp ck4
ck6:    mov byte ptr [si],0         ;nul terminate for field name
        mov ah,15                   ;check monochrome monitor
        int 16
        cmp al,7                    ;if monochrome
        jz ck7                      ;ok
        mov ax,basecolor            ;else
        mov basesc,ax               ;set base color screen address
        mov ax,baseadcolor
        mov basead,ax               ;set color status port
ck7:    call mainrout               ;parameter OK - do main routine
        jmp ck0

mainrout proc near
        call opendbf                ;open DBF file
        jnc mr1                     ;if not success - exit
        ret                         ;invalid dbf file
mr1:    call setfield               ;find field name (second parameter
        jnc mr2                     ;if not found -exit
        ret                         ;invalid field
mr2:    call loadrec                ;load record to buffer
        jnc mr20
        ret                         ;end of file
mr20:   mov si,offset recbuff       ;mr20 to mr5 is to modified
        add si,offfield             ;Thai string (see text)
        mov cx,fieldlen
mr21:   mov al,[si]
        cmp al,'ก'
        jb mr3
        cmp al,'ไ'
        ja mr3
        add al,20
mr22:   mov ah,[si+1]
        mov [si],ah
        mov [si+1],al
        inc si
        dec cx
        jz mr4
        inc si
```

ให้เก็บไว้ในไฟล์ DBF ใหม่ให้ชื่อในที่นี้ว่า <destination> ลำดับขั้นตอนในการทำงานของโปรแกรมของเราจะเป็นดังนี้

1. copy <source> ไปยังไฟล์ใหม่ สมมุติชื่อ <temp> ซึ่งไฟล์นี้จะเหมือนกับ <source> ทุกประการ การก๊อบปี้นี้ อาจใช้คอสก๊อบปี้เพื่อความเร็ว

2. เปลี่ยนแปลงตัวอักษรใน <field> ของ <temp> ให้น้ำหนักถูกต้อง (ดังได้กล่าวไว้แล้วในตอนต้น)

3. ใช้คำสั่งดีเบส use <temp> และทำการ index <field> to <tempdex> จะได้อินเด็กซ์ไฟล์ชื่อ <tempdex> <tempdex> นี้จะเป็นอินเด็กซ์ไฟล์ซึ่งแสดงการเรียงลำดับอักษรตามแบบของภาษาไทยที่ถูกต้องเนื่องจากเราเปลี่ยน <field> ตามข้อ 2 ก่อนทำการ index แล้ว

4. นำเอา <tempdex> ที่ได้ มาใช้กับ <source> โดยคำสั่ง use <source> index <tempdex> อันนี้คือการหลอกดีเบสคือใช้ index ซึ่งได้

รูปที่ 2 (ต่อ)

```
        dec cx
        jz mr4
        jmp mr21
mr3:    cmp al,235
        jz mr5
        cmp al,'.'                  ;chr$(224)
        jb mr31
        cmp al,'.'                  ;chr$(227)
        ja mr31
        mov ah,[si+1]
        cmp ah,'.'                  ;chr$(215)
        jb mr22
        cmp ah,'.'                  ;chr$(221)
        ja mr22
        mov [si],ah
        mov [si+1],al
        inc si
        jmp mr22
mr31:   cmp al,'.'                  ;chr$(221)
        jnz mr32
        mov al,205
        mov [si],al
mr32:   inc si
        dec cx
        jnz mr21
mr4:    call saverec                ;save modified record to file
        inc word ptr reco
        mov ax,reco
        cmp ax,recnum
        ja mr6
        call calcval                ;calculate record number
        mov di,8*160+2*19           ;to be display at line 8 column 19
        call prnval                 ;display
        jmp mr2                     ;goto next record
mr5:    mov di,offset recbuff
        add di,offfield
        add di,fieldlen
        dec di
        mov bx,di
        dec di
mr51:   cmp di,si
        jbe mr52
        mov al,[di]
        mov [bx],al
        dec di
        dec bx
        jmp mr51
mr52:   mov byte ptr [si],205
        mov byte ptr [si+1],'.'     ;chr$(225)
        jmp mr32
mr6:    ret
mainrout endp

opendbf proc near
;this routine open the dBASE file
;check for validity of being valid DBF file
;and load DBF header to headbuff buffer
;headbuff will be as followed
;       offset 0 must be 03h for dBASE III file or 83h for dBASE III plus file
;       offset 4 (double word) = number of records in file
;       offset 8 = offset from begining of file of actual field DATA
;       offset 32 first field name
;              32+10 first field length
;       offset 64 second field name
;              64+10 second field length
;            etc.
;
;opening by this routine will use 2 handles (double opening of the same file)
;first one for reading (will store as DBFHAND)
```

จากการ index file <temp> แต่เอามาใช้กับ <source> ด้วยวิธีนี้ เราจะได้ <field> เดิมที่ไม่มีการเปลี่ยนแปลง แต่มีการเรียงลำดับเรคอร์ด เหมือนกับการเรียงใน <temp> นั่นคือเราสามารถเรียงลำดับอักษรภาษาไทยได้แล้วนั่นเอง

5. copy to <destination> นั่นคือ เราใช้คำสั่งก๊อบปี้ในดีเบสทรีเพื่อก๊อบปี้ไฟล์ที่อินเด็กซ์แล้วนี้ไปยังไฟล์ใหม่ การก๊อบปี้นี้จะก๊อบปี้จาก <source> ทีละเรคอร์ดเรียงตามลำดับใน <tempdex> ไปยัง <destination> เมื่อก๊อบปี้เสร็จก็จะได้ <field> ที่เรียงลำดับอักษรใน <destination> ตามต้องการ

การใช้งานจริง

โปรแกรมสำหรับทำการเรียงลำดับตัวอักษรนี้ ประกอบด้วยโปรแกรม 2 โปรแกรมคือ

1. thaimod.asm ดังโปรแกรมในรูปที่ 2 เป็นโปรแกรมที่เขียนด้วยภาษาแอสเซมบลี หน้าที่ของโปรแกรมนี้คือทำการเปลี่ยนแปลงอักษรใน <field>

รูปที่ 2 (ต่อ)

```
;second one for writing (will store as DBFHAND2)
;by this method of opening will save time not to move file pointer so often
;and speed up file assessment
        mov dx,offset dbfname
        mov ah,openf
        sub al,al                       ;open for read only
        int 21h
        jnc open01
open00: ret
open01: mov bx,ax
        mov dbfhand,ax
        mov dx,offset headbuff
        mov cx,10
        mov ah,readf                    ;read first 10 bytes from DBF file
        int 21h
        cmp cx,ax
        jnz open2
        cmp byte ptr headbuff,3
        jz open44
        cmp byte ptr headbuff,83h
        jz open44
open2:  stc
        ret
open44: call recnum3                    ;set up number of records
        mov si,offset headbuff+8        ;shows offset of begin data
        mov ax,[si]
        mov begdata,ax                  ;save to begdata
        mov cx,[si]
        sub cx,10
        mov dx,offset headbuff+10
        mov ah,readf                    ;read the rest of DBF header
        sub al,al
        int 21h
        cmp cx,ax
        jnz open2
        mov si,offset headbuff
        mov ax,[si+10]
        mov reclen,ax                   ;save record length
        mov word ptr reco,1             ;set record number to 1
        mov bx,dbfhand
        call setpoint                   ;set file pointer to begin of DATA
        mov dx,offset dbfname
        mov ah,openf
        mov al,1                        ;open same file for write only
        int 21h
        jnc open5
        ret
open5:  mov dbfhand2,ax
        mov bx,ax
        call setpoint                   ;set pointer to begin data
        clc
        ret

recnum3: mov si,offset headbuff+6
        mov ax,[si]
        or ax,ax
        jz recnum31
        mov word ptr recnum,0ffffh
        ret
recnum31: sub si,2
recnum21: mov ax,[si]
        mov recnum,ax
        ret
opendbf endp

closedbf proc near
;this routine will close dbf file
        mov bx,dbfhand
        mov ah,closef
        int 21h
```

ให้มีการเรียงน้ำหนักของรหัสแอสกีตามแบบภาษาไทยดังกล่าวข้างต้น และเป็นส่วนที่สำคัญที่สุดของการเรียงคำนี้ โปรแกรมนี้เมื่อทำการคอมพายล์ด้วย masm.exe และ link จะได้ thaimod.exe ซึ่งสามารถรันภายใต้ดอส โดยใช้คำสั่ง

A> thaimod <source.dbf> <field>

แต่ในการใช้งานจริง เราจะทำการเรียก thaimod.exe จากดีเบสโดยตรง เพื่อให้ทำงานได้สะดวกยิ่งขึ้น

แต่เดิมเมื่อ 2-3 ปีก่อน ผู้เขียนเคยเขียนรูทีนโดยใช้ภาษาดีเบสทรีจัดการเรียงน้ำหนักตัวอักษร ซึ่งจะเห็นได้ว่าต้องมีการสลับที่ตัวอักษรค่อนข้างมาก ปรากฏว่าผู้เขียนลองเขียนโปรแกรมในการสลับเฉพาะ เ แ ไ ใ ไ กับอักษรตัวถัดไปเท่านั้น (ยังไม่ได้สลับที่วรรณยุกต์ หรือจัดการ กับ ั เลย) ผลปรากฏว่าโปรแกรมดังกล่าวใช้เวลาในการทำงานมากเหลือเกิน ผู้เขียนทดลองกับไฟล์รายชื่อผู้ป่วย 2500 ราย ซึ่งแต่ละชื่อยาว 35 ไบต์ ทำงานด้วยเครื่องไอบีเอ็มคอมแพตทิเบิ้ล โดยทำงานบนแรมดิสค์

รูปที่ 2 (ต่อ)

```
        mov bx,dbfhand2
        mov ah,closef
        int 21h
        ret
closedbf endp

setpoint proc near
        mov ax,reco
        cmp ax,recnum
        jbe ldrc00
        stc
        ret
ldrc00: dec ax
        mov cx,reclen
        mul cx
        mov cx,dx
        mov dx,ax
        add dx,begdata
        adc cx,0
        mov ah,movep
        sub al,al
        int 21h
        clc
        ret
setpoint endp

setfield proc near
;this routine check field requested in command line
;and see whether the field exist in reqested DBF file
;if the field does not exist then carry will set to show error and exit
;else this routine will store field length to fieldlen
;       offset of field in relation to begining of record to offfield
;fieldlen and offfield will be used by other routine.
        sub dx,dx
        mov word ptr offfield,0     ;offield=0 for first record
setfd5: mov si,offset headbuff+32 ;first record name
        mov di,offset field
        add si,dx
        cmp byte ptr [si],13        ;field search not found
        jz setfd53
setfd521: cmp byte ptr [si],0
        jz setfd54                  ;now field name partly matched
        mov al,[di]
        call ulcase
        mov ah,[si]
        inc si
        inc di
        cmp ah,al
        jz setfd521
setfd522: mov si,offset headbuff+32
        add si,dx
        mov al,[si+16]
        sub ah,ah
        add offfield,ax
        add dx,32
        jmp setfd5
setfd53: stc
        ret
setfd54: cmp byte ptr [di],0
        jnz setfd522                ;not match
setfd542: mov si,offset headbuff+32
        add si,dx
        mov dl,[si+16]
        sub dh,dh
        mov fieldlen,dx
        clc
        ret
setfield endp
```

ต้องใช้เวลาประมาณ 45 นาที กว่าจะจัดการเปลี่ยนอักษรใน <field> ได้หมด เรียกว่ารอกันจนเบื่อทีเดียวละครับ ดังนั้นจึงได้เขียนโปรแกรมด้วยภาษาแอสเซมบลี ดังแสดงในรูปที่ 2 เพื่อจัดการแทน วิธีนี้สามารถลดเวลาการทำงานลงเหลือเพียงประมาณ 30 วินาที คือเร็วขึ้นเกือบ 100 เท่า!

โปรแกรม thaimod.asm ใช้สำหรับจัดน้ำหนักตัวอักษรเพื่อเรียงอักษรภาษาไทยดังได้กล่าวข้างต้น สำหรับผู้ที่ไม่คุ้นเคยกับภาษาแอสเซมบลีอาจอ่านข้ามข้อความต่อไปนี้ได้ ถ้าต้องการใช้โปรแกรม ก็เพียงแต่คีย์ตามนี้ แล้วทำการคอมพายล์ด้วยโปรแกรม masm.exe และ link ด้วยโปรแกรม link.exe ก็จะได้ thaimod.exe ซึ่งสามารถใช้งานได้ครับ

การทำงานโปรแกรม

ส่วนแรกของโปรแกรม (ตั้งแต่ต้นจนถึง ck 7:) จะอ่านที่บรรทัดคำสั่ง ซึ่งต้องประกอบด้วยพารามิเตอร์ 2 ตัว คือชื่อไฟล์ DBF และชื่อฟิลด์ ถ้าไม่พบ

รูปที่ 2 (ต่อ)

```
loadrec proc near
        mov bx,dbfhand
        mov cx,reclen
        mov ah,readf
        mov dx,offset recbuff
        int 21h
        cmp ax,cx
        jae ldrc1
ldrc0:  stc
        ret
ldrc1:  cmp byte ptr recbuff,1ah
        jz ldrc0
        clc
        ret
loadrec endp

saverec proc near
        mov bx,dbfhand2
        mov cx,reclen
        mov ah,writef
        mov dx,offset recbuff
        int 21h
        cmp ax,cx
        jae sdrc1
sdrc0:  stc
        ret
sdrc1:  clc
        ret
saverec endp

calcval proc near
        mov cl,5
        mov bx,offset tempcal
clrval: mov byte ptr [bx],0
        inc bx
        dec cl
        jnz clrval
        mov cx,10
        dec bx
cv0:    xor dx,dx
        cmp ax,10
        jb cv1
        div cx
        add dl,'0'
        mov [bx],dl
        dec bx
        jmp cv0
cv1:    add al,'0'
        mov [bx],al
        ret
prnval: mov bx,offset tempcal
        mov ch,5
        mov cl,4
pv0:    mov al,[bx]
        or al,al
        jnz pv1
        dec ch
        jz pv2
        inc bx
        jmp pv0
pv1:    mov ah,7
        push bx
        push cx
        call poke
        pop cx
        pop bx
        inc bx
        dec cl
        dec ch
        jnz pv0
pv2:    or cl,cl
        jz pv3
        mov al,32
        push bx
        push cx
        mov ah,7
        call poke
        pop cx
        pop bx
        dec cl
        jmp pv2
pv3:    ret
calcval endp

poke proc near
        push es
        cld
        mov dx,basesc
        mov es,dx
        mov dx,basead
        mov bx,ax
        cli
pa:     in al,dx
        test al,1
        jnz pa
pb:     in al,dx
        test al,1
        jz pb
        mov ax,bx
        stosw
        sti
        pop es
        ret
poke endp

ulcase proc near
        cmp al,'z'
        ja ulc0
        cmp al,'a'
        jb ulc0
        sub al,20h
ulc0:   ret
ulcase endp
code ends

data segment
basesc      dw 0b000h
basead      dw 03bah
dbfhand     dw 0
dbfhand2    dw 0
reco        dw 0
recnum      dw 0
reclen      dw 0
begdata     dw 0
tempcal     db 6 dup (0)
dbfname     db 65 dup (0)
field       db 11 dup (0)
offfield    dw 0
fieldlen    dw 0
basecolor   dw 0b800h
baseadcolor dw 03dah
headbuff    db 2000 dup (?)
recbuff     db 4000 dup (?)
data ends
 end
```

พารามิเตอร์ดังกล่าวในบรรทัดคำสั่งก็จะหยุดการทำงาน แต่ถ้าพบพารามิเตอร์ทั้ง 2 นี้ โปรแกรมจะทำงานต่อไป

mainrout เป็นเมนรูทีนของโปรแกรม ซึ่งจะเริ่มทำงานโดยเปิดไฟล์ DBF (โดย opendbf routine) ตรวจสอบฟิลด์ที่ต้องการ (โดย setfield routine) ถ้าพบความผิดพลาด คือไม่พบไฟล์ หรือไฟล์ที่พบไม่ใช่ไฟล์ดีเบส หรือไม่พบฟิลด์ที่สั่ง โปรแกรมก็จะหยุดทำงาน

ถ้าทุกอย่างถูกต้อง mainrout จะอ่านเรคอร์ดจากไฟล์ DBF ลงมาในบัฟเฟอร์ทีละเรคอร์ด (โดย loadrec routine) และทำการเปลี่ยนแปลงอักษรของฟิลด์ที่กำหนด โดยถือหลักที่กล่าวมาแล้วข้างต้น (รูทีนตั้งแต่ mr20: ถึง mr4:, mr5: ถึง mr6:) เมื่อเปลี่ยนแปลงเสร็จสิ้นจะเขียนกลับลงไปที่เดิม (โดย saverce routine) หลังจากนั้นจะทำงานกับเรคอร์ดต่อไปจนสิ้นสุดไฟล์ DBF ขณะที่ทำงานนี้ โปรแกรมจะแสดงหมายเลขเรคอร์ดบนจอภาพด้วย (โดย calcval และ prnval routine)

opendbf ทำการเปิดไฟล์ DBF อ่าน header ของไฟล์ลงยัง headbuff ใน headbuff นี้จะบอกถึงโครงสร้างของไฟล์ DBF นั้น บอกถึงฟิลด์รวมทั้งความยาวของฟิลด์นั้นๆด้วย การเปิดไฟล์นี้ใช้แฮนเดิล 2 ตัวคือ เปิดไฟล์เดียว 2 ครั้ง โดยแฮนเดิลตัวหนึ่งสำหรับอ่านไฟล์และอีกตัวหนึ่งสำหรับเขียนไฟล์ วิธีนี้ทำให้อ่านเขียนไฟล์ได้โดยเร็ว เพราะไม่ต้องเลื่อนไฟล์พอยน์เตอร์บ่อยๆ หลังจากเปิดไฟล์

setfield ทำการตรวจสอบว่าชื่อฟิลด์ที่ส่งมาในบรรทัดคำสั่งมีอยู่หรือไม่ ถ้ามีอยู่จะตรวจสอบความยาว และออฟเซตของฟิลด์นั้นในเรคอร์ด

loadrec ทำการโหลดเรคอร์ดลงสู่บัฟเฟอร์ครั้งละเรคอร์ด

saverec ทำการเซฟเรคอร์ดจากบัฟเฟอร์ครั้งละเรคอร์ดลงดิสค์

สำหรับรูทีนอื่นๆ เป็นรูทีนซัพพอร์ต เช่นใช้แสดงผลทางจอภาพ ใช้แสดงตัวเลขเรคอร์ดขณะทำงาน

2. thaisort.prg เป็นโปรแกรมหลักสำหรับการเรียงลำดับอักษร เขียนด้วยภาษาดีเบสทรี แสดงไว้ในโปรแกรมรูปที่ 3 การทำงานของโปรแกรมนี้ จะเป็นไปตามแนวความคิดข้างต้น โดยโปรแกรมนี้จะเรียก thaimod.exe มาใช้เองโดยอัตโนมัติ

การใช้งานโปรแกรมนี้ อาจใช้กับคลิปเปอร์คอมพายเลอร์ โดยผู้ใช้คีย์โปรแกรมนี้ แล้วคอมพายล์ด้วยคลิปเปอร์คอมพายเลอร์ (ผู้เขียนใช้เวอร์ชัน Winter 85) และ link ด้วย plink86 หรือ DOS linker จะได้ thaisort.exe ซึ่งสามารถรันภายใต้คอสได้เลย ในการรันวิธีนี้ จะต้องมีโปรแกรม thaimod.exe ข้างต้น อยู่ใน logged drive และรันด้วยคำสั่ง

THAISORT <source file name> on <field name> to <destination file name>'

<source file name> คือชื่อไฟล์ DBF ที่ต้องการเรียงลำดับอักษร

<field name> คือชื่อฟิลด์ที่มีภาษาไทยที่ต้องการเรียงลำดับอักษรอยู่

<destination file name> คือชื่อไฟล์ที่ได้หลังจากเรียงลำดับเสร็จสิ้นแล้ว

ตัวอย่างเช่น เราต้องการเรียงลำดับอักษรในไฟล์ ชื่อ CUSTOMER.DBF โดยมีฟิลด์ชื่อ NAME และเมื่อเรียงเสร็จแล้วให้ได้ไฟล์ชื่อ B:CUSTSORT.DBF เราใช้คำสั่งดังนี้

A> THAISORT CUSTOMER ON NAME TO B:CUSTSORT

หลังทำงานเสร็จสิ้น ก็จะได้ไฟล์ CUSTSORT.DBF ซึ่งมีการเรียงลำดับอักษรในฟิลด์ NAME

อย่าลืมว่า ใน logged drive จะต้องมีที่เหลือพอ สำหรับอินเด็กซ์ไฟล์ด้วย

ถ้าต้องการความเร็วสูงสุด อาจทำงานในแรมดิสค์โดยเปลี่ยน logged drive เป็นแรมดิสค์ (สมมุติว่าเป็นไดรฟ์ C:) ดังนี้

รูปที่ 3 โปรแกรม Thaisort.prg.

```
*************************
* LISTING THAISORT.PRG *
*************************
* by Choosna Makarasara MD.
* Rajavithi Hospital Micro Computer Club
* 6 February 1988
* This program is for sorting thai language
* stored in DBF file      .

PARAMETER db
set intensity off
set talk off
set safety off
udb=trim(upper(db))
if .not. ('ON'$udb .and. 'TO'$udb)
  ? 'invalid parameter'
  ?
  ? 'using THAISORT <filename> on <fieldname> to <ntx filename>'
  ?
  return
endif

aton=at('ON',udb)
atto=at('TO',udb)

filename=trim(subs(udb,1,aton-1))
file2=filename
if .not. '.'$filename
   file2=filename+'.DBF'
```

```
else
   filename=subs(filename,1,at('.',filename))
endif
if .not. file("&file2")
  ? 'file '+filename+' does not exist.'
  ?
  return
endif
use &filename
field=trim(subs(udb,aton+3,atto-aton-4))
if type("&field")="U"
  ? 'field '+field+' does not exist.'
  ?
  return
endif
use

clear
col=(79-len(trim(db))-8)/2
@ 1,col say 'SORTING '+db

slave=trim(subs(udb,atto+3,len(udb)-atto-2))
if '.'$slave
   slave=subs(slave,1,at('.',slave))
endif
slave2=slave+'.DBF'

@ 4,0 say "Copying &file2 to &slave2"
?
run copy &file2 &slave2
@ 7,0 say 'Modifying file &slave2 '
@ 8,10 say 'RECORD : '
run thaimod &slave2 &field

set talk on
use &slave
@ 10,0 say 'Indexing '+slave+' on '+field+' to '+"TEMPFILE.NTX"

index on &field to tempfile.ntx

use &filename index tempfile.ntx
@ 13,0 say 'Copy sorted file to &slave2'
copy to &slave
use
set talk off
dele file tempfile.ntx
set talk on
@ 16,0 say 'COMPLETED '
? chr(7)
return
```

C > A : THAISORT CUSTOMER ON NAME TO B : CUSTSORT

ในกรณีนี้หมายความว่า โปรแกรม thaisort.exe อยู่ในไดรฟ์ A: และได้ก๊อบปี้ CUSTOMER.DBF มาอยู่ในไดรฟ์ C: และก่อนรันวิธีนี้ต้องก๊อบปี้ thaimod.exe ไว้ในไดรฟ์ C: เสียก่อนด้วย เมื่อสิ้นสุดการทำงานจะได้ CUSTSORT.DBF ในไดรฟ์ B:

ในกรณีที่บรรทัดคำสั่งไม่ถูกต้อง เช่นไม่พบไฟล์ หรือชื่อฟิลด์ไม่ถูกต้อง จะมีการแสดง error บนจอ

โปรแกรมนี้ ถ้ารันภายใต้ดีเบสทรีพลัสหรือ FOX plus ให้ใช้คำสั่ง DO THISORT WITH '<source file name> on <field name> to <destination file name>

รายละเอียดอื่นๆคล้ายกับที่กล่าวมาแล้วข้างต้น

การทดสอบ

ผู้เขียนได้ทำการทดสอบโปรแกรมดังกล่าวนี้ โดยฮาร์ดแวร์ประกอบด้วยพีซีเอ็กซ์ทีเทอร์โบ 8 MHz ทำงานภายใต้ฮาร์ดิสก์ขนาด 20 เมกะไบต์ ซึ่งมีความเร็วเฉลี่ยในการเข้าถึงข้อมูลประมาณ 65 msec ทดลองเรียงอักษรในไฟล์ DBF โดยเรียงชื่อฟิลด์ยาว 35 ไบต์ และเรคอร์ดยาว 69 ไบต์ จำนวนเรคอร์ดทั้งหมด 2596 เรคอร์ด ได้ผลดังต่อไปนี้

รันโปรแกรมซึ่งคอมพายล์โดยคลิปเปอร์ ใช้เวลา 4 นาที 57 วินาที

รันด้วยดีเบสทรี ใช้เวลา 6 นาที 33 วินาที

รันด้วย Foxplus เวอร์ชัน 2.0 ใช้เวลา 1 นาที 40 วินาที

จะเห็นได้ว่า การรันภายใต้ Foxplus ให้ความเร็วสูงสุด ทั้งนี้เป็นเพราะ Foxplus มีรูทีนในการอินเด็กซ์และก๊อบปี้ไฟล์ที่ดี ส่วนคลิปเปอร์นั้น แม้ว่าจะเป็นภาษาที่คอมพายล์แล้ว แต่รูทีนในการอินเด็กซ์และก๊อบปี้ยังไม่ดีนัก ทำให้เสียเวลามากในส่วนนี้ สำหรับดีเบสทรีนั้น รันได้ช้าที่สุด และถ้าจำนวนเรคอร์ดยิ่งมากขึ้นจะยิ่งมีปัญหาเรื่องความล่าช้ามากขึ้นด้วย เพราะดูเหมือนเรคอร์ดหลังๆจะยิ่งช้าลงเรื่อยๆ สำหรับดีเบสทรี

โดยสรุปก็คือ การเรียงลำดับอักษรภาษาไทยนั้นสามารถทำได้ และรูทีนที่เขียนมานี้ ผู้เขียนหวังว่าจะเป็นประโยชน์ไม่มากก็น้อย โปรแกรมนี้ เป็นความพยายามอันหนึ่งที่จะเสริมจุดด้อยของดีเบสทรีในเรื่องความล่าช้าในการจัดการกับไฟล์ โดยนำเอาภาษาแอสเซมบลีเข้ามาช่วยให้การทำงานเร็วยิ่งขึ้น ผู้เขียนเชื่อว่ารูทีนเหล่านี้จะมีประโยชน์ และสามารถนำไปดัดแปลงและประยุกต์ใช้กับดีเบสทรีภายใต้สถานะการอื่นๆได้อีก เพื่อเพิ่มสมรรถนะของดีเบสทรีให้ดียิ่งขึ้น

สักวันหนึ่งในอนาคต ถ้าเรามีตัวจัดการฐานข้อมูลของเราเองซึ่งมีความเป็นไทยมากกว่าไปก๊อบปี้โปรแกรมของต่างประเทศใช้ เราคงสามารถเรียงลำดับอักษรภาษาไทยได้ง่ายกว่าเดิม และคงจะใช้งานได้สะดวกขึ้นกว่าการดัดแปลงโปรแกรมต่างประเทศดังกล่าว ขณะนี้คงเป็นเพียงความฝันไปก่อนนะครับ